IET Networks

Special Issue: Edge Network Computing for Big Data Systems over Internet of Things

IET Journals
The Institution of Engineering and Technology

Deep learning approach to multimedia traffic classification based on QoS characteristics

ISSN 2047-4954
Received on 7th October 2018
Revised 14th November 2018
Accepted on 7th December 2018
E-First on 14th January 2019
doi: 10.1049/iet-net.2018.5179
www.ietdl.org

Zaijian Wang[1] ✉, *Shiwen Mao*[2], *Weidong Yang*[3]

[1]*College of Physics and Electronic Information, Anhui Normal University, Wuhu, Anhui 241000, People's Republic of China*

[2]*Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA*

[3]*Key Laboratory of Grain Information Processing and Control, Henan University of Technology and Ministry of Education, Zhengzhou, Henan 450001, People's Republic of China*

✉ *E-mail: wangzaijian@ustc.edu*

Abstract: With the fast increase of multimedia traffic in Internet of Things (IoT) applications, IoT traffic now requires very different Quality of Service (QoS). By extensive statistical analysis of traffic flow data from a real world network, the authors find that there are some latent features hidden in the multimedia data, which can be useful for accurately differentiating multimedia traffic flows from the QoS perspective. Under limited training data conditions, existing shallow classification methods are limited in performance, and are thus not effective in classifying emerging multimedia traffic types, which have truly entered the era of big data and become very completed in QoS features. This situation inspires us to revisit the multimedia traffic classification problem with a deep learning (DL) approach. In this study, an improved DL-based multimedia traffic classification method is proposed, which considers the inherent structure of QoS features in multimedia data. An improved stacked autoencoder model is employed to learn the relevant QoS features of multimedia traffic. Extensive experimental studies with multimedia datasets captured from a campus network demonstrate the effectiveness of the proposed method over six benchmark schemes.

Nomenclature

K	denote the total output value number of corresponding QoS characteristics
x_k	denote the output value of QoS characteristics
$\boldsymbol{X}$	represent a set including all possible outputs
$I(x_k)$	denote the information content of x_k
$p(x_k)$	denote the probability density function of x_k
N	denote the number of inputs
D	denote the number of units in the output layer
H_D	denote the number of hidden units
$\boldsymbol{M}$	denote a matrix that consists of the weight vectors
R^d	represent an N-dimensional real number Euclidian space
$x^{(i)}$	specify the ith training sample
$F(\cdot)$	denote a feature-extracting function
$\boldsymbol{W}_1$	represent an encoder weight matrix
$\boldsymbol{b}$	represent an encoder bias vector
$\boldsymbol{W}_2$	denote a decoder weight matrix
$\boldsymbol{c}$	denote a decoder bias vector
γ	denote the weight
ρ	represent a probability and equals to a small value close to zero (set to 0.005 in this paper)
$\hat{\rho}_j$	represent the average activation of hidden unit j

1 Introduction

In Internet of Things (IoT), multimedia communications have gained great momentum in recent years. For example, in video surveillance and smart homes, smartphones/TVs are able to communicate with each other via heterogeneous wireless networks (e.g. WiFi and 4G) [1–3]. More people have paid attention to research on Quality of Service (QoS) for multimedia communications in the IoT [4–6], in contrast to traditional wireless networks [7–10], where various models are developed to facilitate resource allocation and traffic control and network protocols developed for end-to-end QoS provisioning. To effectively guarantee end-to-end QoS for multimedia applications, it is important to obtain accurate QoS information of traffic flows, which will be helpful information for Internet Service Providers (ISPs) to make better QoS operation decisions. Especially, it is a great challenge to effectively allocate limited network resources for emerging multimedia traffics in future generation of networks (e.g. the fifth generation wireless networks (5G)), that are becoming increasingly 'big' and have many different QoS requirements since they cannot be handled effectively by the traditional multimedia processing and analysis methods, in spite of that big data can bring great opportunities [11–13].

With the proliferation of multimedia applications (such as multi-view video, interactive video systems, real-time surveillance, and 3D and 360° videos), multimedia has become the 'biggest big data' and can offer rich, important information for service and network operators [11, 12]. Compared with big data, multimedia big data is much more complex, which involves more QoS factors since many multimedia traffic flows need more strict QoS requirements. Multimedia big data is composed of partially unknown complex structures, which is difficult to represent and model. Especially some factors involve time statistics, spatial statistics, human factors, and inter-view correlations with structured singularities. Furthermore, there is a higher level of complexity involved in understanding and cognition of multimedia big data since network parameters usually do not reflect high-level semantics [11]. Therefore, multimedia big data can offer more hidden characteristics than traditional big data. In the field of end-to-end QoS for multimedia traffic, more effective potential patterns may be found by utilising multimedia big data. To differentiate multimedia traffic flows at different priority levels for effective end-to-end QoS guarantee, it is important to analyse QoS characteristics of multimedia traffic from the perspective of multimedia big data.

In this paper, a deep learning (DL) approach is utilised to learn the hidden patterns of QoS characteristics from unknown complex structures of multimedia big data. We focus on multimedia traffic classification, with the objective to extract such QoS information from captured multimedia data, according to which different multimedia traffics can be differentiated into different priority levels. Traffic classification has gained more attention in the research community, with the rapid development of network traffic

and user's individual requirements. However, typical classification methods are not effective for multimedia big data due to several reasons as follows:

i. They mainly depend on acquired traffic data, and cannot identify the inherent features hidden in the unprecedented volumes of non-traditional data.
ii. They often classify traffic based on linear division without considering the fact that the related QoS features exhibit great diversity in type and complexity, as well as the relationship among various parameters.
iii. They are often built on top of direct features, which are often much restrictive in capturing the complex multimedia QoS features with domain-specific knowledge and large spatial variability.
iv. The volume of multimedia traffic is growing rapidly, in which considerable potential information may be hidden and needs to be effectively extracted [14].

The recent developments on DL open an exciting new era in pattern recognition and machine learning [14–19], with wide applications in multimedia computing and communications [20–22]. DL offers great potential to extract the inherent QoS related features of multimedia data and discover the QoS feature structure without the need for prior knowledge, which are usually abstract and invariant. It has been recognised that the machine learning methods with multiple layers of processing can yield higher classification accuracy than those traditional, shallower classifiers [23]. In this paper, we introduce a DL-based QoS related feature extraction scheme for multimedia traffic classification. This paper is focused on utilising the autoencoder (AE) to learn the QoS related features of multimedia traffic with an unsupervised learning approach. Our main contributions include

i. To the best of our knowledge, this is the first work on exploiting multimedia traffic QoS related features for classification, with a deep architecture model in which AEs are used as building blocks.
ii. To enhance the learning process and to reduce the training error caused by a very small number of training patterns, we improve the architecture of the AE and propose a modified DL approach of multimedia data from the QoS perspective.
iii. We validate the performance of the proposed DL approach with extensive multimedia data captured from a campus network over a long period of time, and with comparison to six existing representative classification schemes. The proposed method is shown to outperform all the six existing schemes with considerable gains with respect to all five performance metrics.

The remaining of this paper is organised as follows. Section 2 presents the related work in the literature. Section 3 introduces the dataset and analyses the related typical QoS characteristics, and selects new QoS characteristics for differentiating multimedia traffics. Section 4 presents an improved DL approach to multimedia big data. Experimental results are presented and discussed in Section 5. Finally, Section 6 concludes this paper.

2 Related work

With the increasing popularity for multimedia services in the IoT, multimedia communications play an increasingly larger role in the IoT applications. Research on QoS in multimedia communications in the IoT has gained attention in recent years [1–6]. In [2], the authors propose a novel vehicle network architecture in the smart city scenario to deliver delay-tolerant and delay-sensitive traffic requiring very different QoS. To ensure the quality of multimedia applications, Karaadi *et al.* [3] design a quality aware IoT architecture for multimedia IoT applications. Addressing the effect in heterogeneity of applications, services, and terminal devices, and the related QoS issues among them, the work [4] proposes a traffic flow management policy. To allocate cache capacity among content-centric computing nodes and handle the transmission rates under a constrained total network cost, the authors in [5] propose a suboptimal dynamic approach, which is suitable for the IoT with frequently content delivery. Furthermore, an IoT-based architecture is proposed in [6] for multi-sensorial media delivery to TV users in a home entertainment scenario, and Song and Tjondronegoro [24] utilises statistical non-linear regression analysis to build the models with a group of influencing factors as independent predictors, which include encoding parameters, bitrate, video content characteristics, and mobile device display resolution.

Big data attracted an upsurge of research recently, which may provide the great potential to obtain valuable knowledge from the exponential growth and wide availability of non-traditional data [14]. Big data has been defined by volume, velocity, veracity, and variety, which indicate that the data not only has a large data measure, but also has different modalities and types for a given object [12, 13]. The emergence of IoT-based multimedia has challenged many of the traditional QoS feature analytic methods. Comprehensive surveys of big data can be found in [12, 13]. The authors in [25] show that accurate and timely traffic flow information is helpful in improving traffic operation efficiency, and proposed a novel traffic flow prediction method based on deep architecture models with big traffic data. In [26], the authors develop a novel community-centric framework for community activity prediction based on big data analysis.

DL, which is a type of machine learning method and tries to hierarchically learn deep features of input data with very deep neural networks (NNs), has drawn a lot of academic and industrial interest [27, 28]. In DL algorithms, multiple-layer architectures or deep architectures are adopted to extract inherent features in data from the lowest level to the highest level, by which huge amounts of potential structure can be discovered in data and proper features can be formulated for pattern classification in the end. Since deep models can potentially lead to progressively more abstract and complex features at higher layers, deep models can give a better approximation to non-linear functions than shallow models because more abstract features are generally invariant to most local changes of the input. Generally, deep belief networks (DBNs), deep Boltzmann machines (DBMs), SAEs, and stacked denoising AEs (SDAEs) are the typical deep NN architectures. Furthermore, the layer-wise training models have a bunch of alternatives such as denoising AEs (DAE), convolutional NNs (CNNs), pooling units, AEs, and restricted Boltzmann machines (RBMs).

DL and multimedia big data are both active and interdisciplinary research hotspots [11]. DL can learn high-level features from low-level ones with a deep NN, which can exploit the proper features for classification [29]. Since multimedia big data could offer a great potential to obtain valuable knowledge, the authors in [14] show that DL is playing a key role in providing big data predictive analytic solutions, and has also been successfully applied in big data analytics. In [15–19], DL is shown highly effective for handing the channel state information (CSI) data for indoor localisation. For video sequence classification, the authors in [30] propose a novel tensor decomposition method in which general tensors are used as input for video sequence classification. The method projects the original tensor into subspaces spanned by spatial basis matrices in the proposed formulation. By exploring the motion relationship of neighbouring blocks and the coding cost characteristic, Fan *et al.* [31] categorise prediction unit (PU) into one of three classes, namely, motion-smooth PU, motion-medium PU and motion-complex PU. By exploiting traffic patterns and Variable Bit-Rate encoding, Dubin *et al.* [32] present a new algorithm for encrypted HTTP adaptive video streaming title classification. The algorithm shows that an external attacker can identify the video title from video HTTP adaptive streams sites, such as YouTube. By training a DBN, the work in [33] defined a deep architecture for traffic flow prediction that learned features with limited prior knowledge. In [34], a discriminative deep model was trained to classify the features in a blind image quality assessment model. In [35], a DL scheme was proposed to infer possible diseases. To effectively model the interaction relationships, the deep NNs and a latent structural support vector machine (SVM) were jointly utilised in [36], while in [37], the

Table 1 Sixteen types of related QoS characteristics

No.	Name	Description Bayes
1	downlink-uplink-packets-count-ratio	ratio of packet number between downlink and uplink
2	downlink-uplink-bytes-count-ratio	ratio of byte number between downlink and uplink
3	downlink-subflow-count	number of sub-flows from downlink
4	downlink-different-ipcount	number of different IP addresses from downlink
5	downlink-rate-packet-pdf-entropy	entropy of PDF of packet rate from downlink
6	downlink-rate-packet-big-pdf-entropy	entropy of PDF of big packet rate from downlink
7	downlink-rate-bytes-pdf-entropy	entropy of PDF of byte rate from downlink
8	downlink-rate-bytes-big-pdf-entropy	entropy of PDF of big byte rate from downlink
9	downlink-packetsize-pdf-entropy	entropy of PDF of packet size from downlink
10	downlink-ipchange-count-pdf-entropy	entropy of PDF for the count computed by changing between different IP addresses
11	downlink-interval-pdf-entropy	entropy of PDF of packet arrival time interval from downlink
12	downlink-flow-segment-time-pdf-entropy	entropy of PDF of packet arrival time interval from downlink
13	downlink-flow-segment-rate-pdf-entropy	entropy of PDF of flow segment rate from downlink
14	downlink-flow-segment-interval-pdf-entropy	entropy of PDF of flow segment arrival time interval from downlink
15	uplink-packets-count	number of packets from uplink
16	uplink-bytes-count	number of bytes from uplink

authors built adversaries for DL systems applied to image object recognition.

Multimedia big data provide unprecedented opportunities for understanding real-world situations [11]. However, it is an interdisciplinary research field to integrate machine learning (e.g. DL), big data, multimedia traffic modelling/analysis/control, and QoS feature together. Considering that DL can exploit the proper features for classification since it can learn high-level features from low-level ones with a deep NN, we choose SAEs as the corresponding deep architecture in this paper [15–17, 25, 38, 39]. Our work presented in this paper makes a step forward in applying DL for video traffic classification in this important topic area.

3 Dataset analysis

To further study the related QoS characteristics of multimedia traffics, we collect the typical multimedia traffic traces in a college campus network to build a basic dataset with Wireshark, a widely-used network protocol analyser [40]. The dataset is comprised with eight types of multimedia traffic flows, including video based on http, PPstream video, QQ video, sopcast video, CCTV online video, xunlei video, youku standard-definition video (youku-b), and youku high-definition video (youku-g); (The websites for these multimedia applications are http://xyq.163.com/, http://www.QQ.com, http://www.cntv.cn/, http://www.uusee.com/, PPStream is a network television software, http://dl.xunlei.com/, http://www.sopcast.cn, and http://www.youku.com/.) each traffic flow is 35 min long, collected with a machine with an AMD A6-7000 Radeom processor during day/night times in the summer of 2015.

When we collected each type of the multimedia traffic traces, we kept only one type of multimedia traffic running without other types multimedia traffic in the background. Note that this is a typical scenario when a single user is viewing a video on his/her desktop computer (i.e. usually a user is not viewing multiple videos simultaneously). However, there may be other users viewing multimedia contents on other computers in the lab or other parts of the campus network, while the multimedia traffic trace is recorded. Wireshark recorded detailed information of the multimedia traffic flows, such as the five-tuples (including, source/destination IP address, source/destination port, and protocol), the arrival time, the direction, the packet size and so on. We consider both uplink and downlink directions of the traffic data.

For better exposition in the remainder of this paper, we first provide the following definitions. The notation used in the remaining part of this paper is summarised in the Nomenclature section.

- *Sub-flow*: a sub-flow consists of packets with the same five-tuples and from the same application. Its duration should be longer than 0.1 s.
- *pdf-entropy*: the entropy of the probability density function (PDF) is defined as follows:

$$H(\boldsymbol{X}) = \mathbb{E}[I(\boldsymbol{X})] = -\sum_{k=1}^{K} p(x_k)\log_2 p(x_k), \tag{1}$$

  where K denotes the total output value number of the corresponding QoS characteristics, x_k denotes the kth output value of the QoS characteristics, $\boldsymbol{X} = [x_1, x_2, \ldots, x_k, \ldots, x_K]$ represents a set including all possible outputs, $I(x_k)$ is the information content of $\boldsymbol{X}$, and $p(x_k)$ denotes the PDF of x_k.
- *packet-big*: a packet is marked as 'big' if its size is larger than 800 Bytes.
- *ipchange-count-pdf-entropy*: the entropy of PDF for the count computed by changing between different IP addresses.
- *flow segment*: indicates each entire sub-flow.

For multimedia traffic classification, feature plays an important role and can greatly affect classification performance. In practice, since the QoS requirements of multimedia traffic are complicated in nature, QoS feature extracted should be more invariant and robust to most local changes of the input. Generally, the features at a lower level have poor classification performance since it is too simple; the features at a higher level may have the ability of acquiring a better classification performance since the deeper features can preserve abstract and invariant information. We find that the eight types of multimedia traffic flows cannot be effectively classified with simple relevant QoS features, such as protocol, port, packet size, Inter-Arrival Time (IAT) and so on. However, it is possible to discover the more abstract inherent feature structures in a huge amount of simple related QoS features. By analysing related works, we select 16 types of related QoS characteristics, which are presented in Table 1, to describe the captured multimedia traffic traces.

To better understand all the captured traffic traces, the normalised logarithmic values of all related characteristics are plotted in Figs. 1–3, respectively. For example, in Fig. 1*a*, the *x*-axis is the normalised logarithmic value of uplink-bytes-count, and the *y*-axis is the normalised logarithmic value of the downlink-rate-packet-big-PDF-entropy. To make the figures more readable, we plot only one out of every ten points in the figures. From Figs. 1–3, the distributions of normalised logarithmic value of the related characteristics can be compared easily for the captured traffics.

As shown in Fig. 1*a*, the entire distribution area cannot be clearly divided into subareas each containing one types of points, since all of traffic traces are out of order and mingled, and cannot be clearly distinguished from each other with the two features (i.e. the normalised logarithmic value of uplink-bytes-count and downlink-rate-packet-big-pdf-entropy). In particular, 'http' is

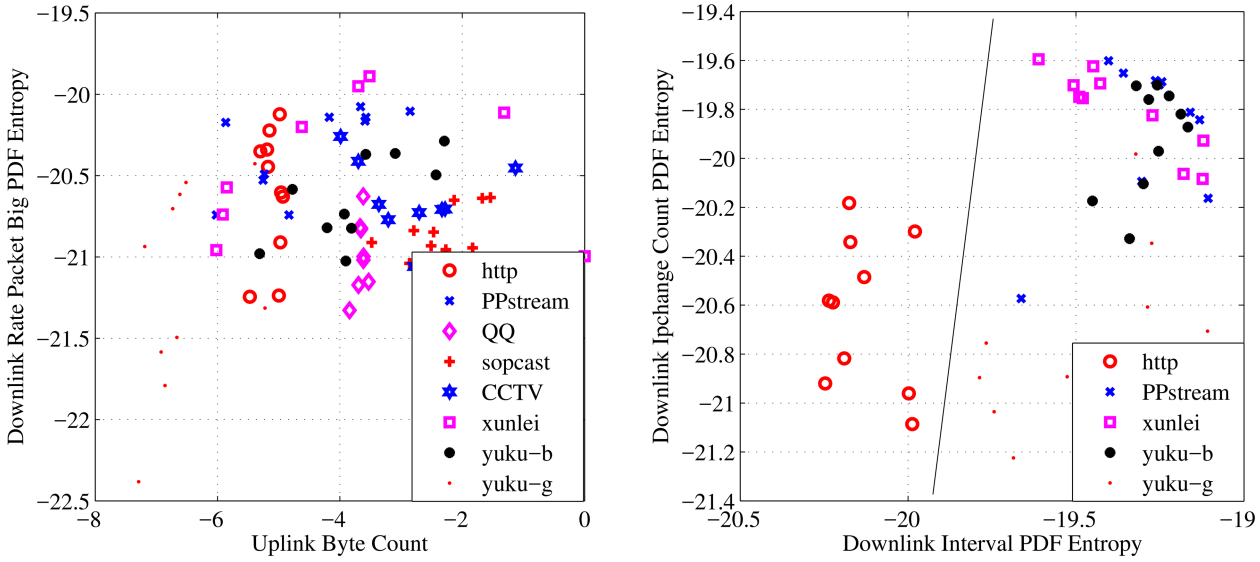

Fig. 1 *Distributions of normalised logarithmic values of related QoS characteristics for eight types of traffic flows*

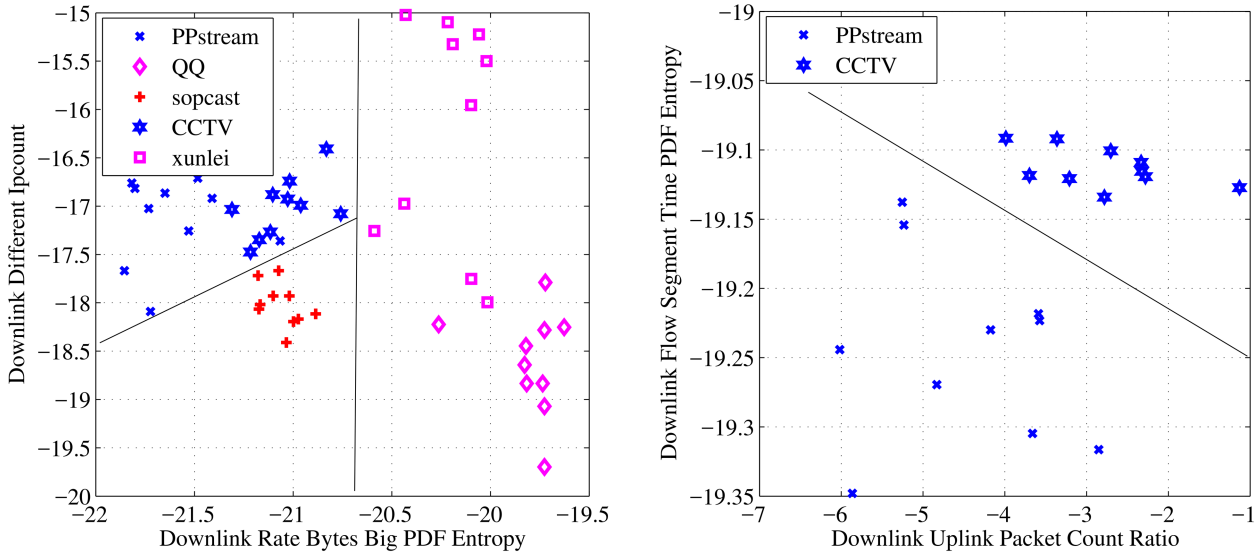

Fig. 2 *Distributions of normalised logarithmic value of related QoS characteristics for five types traffic flows*

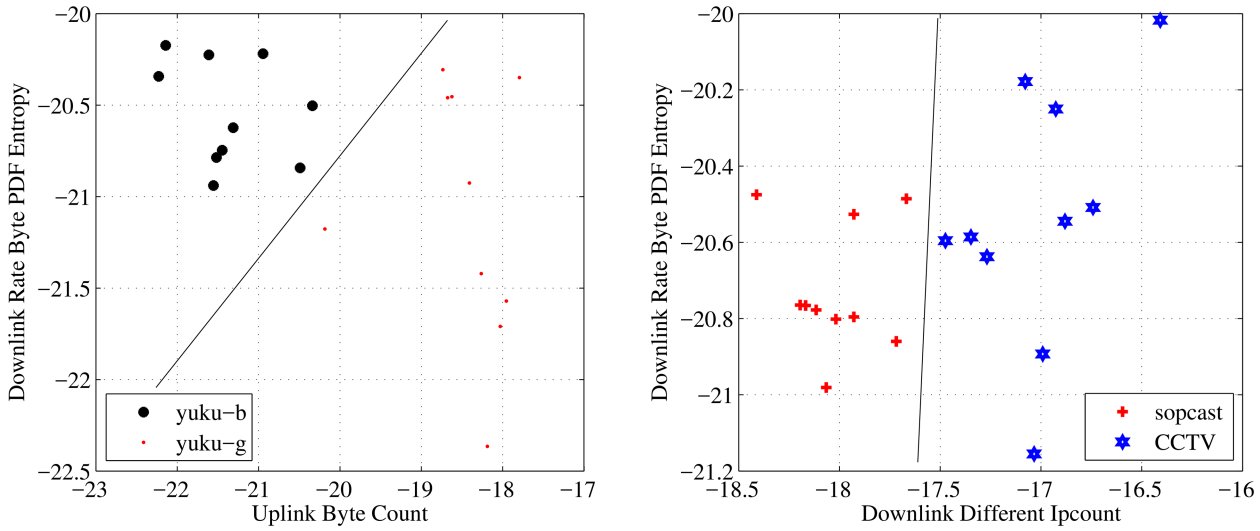

Fig. 3 *Distribution of normalised logarithmic values of related QoS characteristics for four types traffic traces*

mixed with four traffic traces (i.e. 'PPstream,' 'xunlei,' youku-b, and youku-g) in Fig. 1*a*; 'QQ' is mixed with three traffic traces ('sopcast,' 'CCTV,' and youku-b) in Fig. 1*a*; and 'xunlei' is mixed with five traffic traces ('http,' 'PPstream,' 'sopcast,' 'CCTV,' youku-b, and youku-g), as shown in Fig. 1*a*.

However, it is obvious that 'http' can be clearly distinguished from the others as shown in Fig. 1*b*. In Fig. 1*b*, 'http' has the

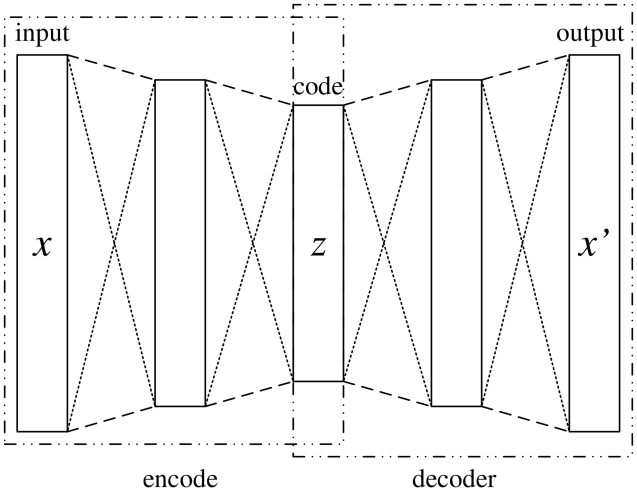

Fig. 4 *Schematic structure of an AE with three fully-connected hidden layers*

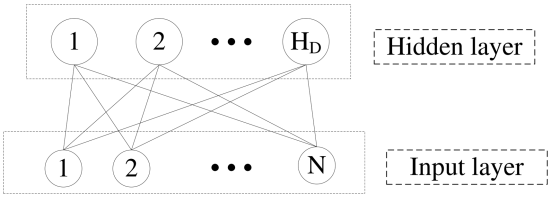

Fig. 5 *Input-to-hidden layer structure of an AE*

smallest value among all with respect to downlink interval pdf entropy, i.e. its distribution of downlink interval pdf entropy is obviously different from 'PPstream,' 'xunlei,' youku-b, and youku-g. Nevertheless, 'PPstream,' 'xunlei,' youku-b, and youku-g can be distinguished from each other according to the distribution of downlink ipchange count PDF entropy and downlink interval PDF entropy.

From the distribution of normalised logarithmic value of downlink different ipcount and downlink rate bytes big PDF entropy as shown in Fig. 2*a*, 'xunlei' and 'QQ' can be easily distinguished among the five types traffic traces (i.e. from 'PPstream,' 'sopcast,' and 'CCTV'), and 'Sopcast' can be easily distinguished from 'PPstream' and 'CCTV.' In Fig. 2*b*, 'PPstream' has obvious differences from 'CCTV' in the distribution of normalised logarithmic value of downlink uplink packets count ratio and downlink flow segment time PDF entropy.

From the distribution of normalised logarithmic value of downlink rate bytes PDF entropy and uplink bytes count, as shown in Fig. 3*a*, youku-b exhibits obvious differences from youku-g. In Fig. 3*b*, 'sopcast' can be easily distinguished from 'CCTV' based on the distribution of normalised logarithmic value of downlink different ipcount and downlink rate bytes PDF entropy.

In conclusion, it is very different to distinguish the eight traffic traces according to just one feature. But different traffic flows have different features with respect to different related QoS characteristics. This observation indicates that related QoS characteristics should have some potential relationships among them, by which the traffic flows can be effectively distinguished. Multimedia big data could provide more potential information to find the inherent structure for multimedia traffic classification.

4 DL approach to multimedia big data

In this paper, a promising classification method, a DL approach to multimedia big data based on QoS characteristics (termed DeepClass), is introduced for multimedia traffic QoS classification. The proposed method exploits a stacked AE (SAE) model, which creates a deep network by utilising AEs as building blocks. It has been shown that the SAE model can extract different levels of potential classification features in a broad area of applications and can discover a huge amounts of inherent structures in the features without prior knowledge [25]. We describe how to incorporate the SAE model into a traffic related QoS feature classification framework, which is complicated in nature to be represented.

4.1 Autoencoder

The AE is typically implemented as a one-hidden layer NN, which attempts to reproduce its input. It is used as building blocks to train deep networks, where the target output is the input of the model and each hidden layer is associated with an AE that can be trained separately. The amount of output nodes is equal to the amount of input nodes, and one input node is trained at a time. In general, an AE has one input layer, one hidden layer, and one output layer [15–17, 25, 38, 39].

An AE is composed with four parts as follows: (i) the input layer, (ii) the output layer, (iii) one or more hidden layers, and (iv) an activation function [38]. The schematic structure of an AE with three fully-connected hidden layers is shown in Fig. 4. The input layer is visible with N inputs, and the output layer is a reconstruction layer with D units. In our design, the hidden layer is invisible with H_D units. That is, we will utilise an AE with N input neurons and H_D hidden neurons. Each of the hidden units is connected with each of input neurons in the input-to-hidden layer of an AE (i.e. fully connected), as shown in Fig. 5. Therefore, every single hidden unit has N connections from the input layer, which as a whole both filter away information from some input data and amplifies some others. Therefore, we can regard the learning process of an AE with H_D hidden units as a learning process with H_D such filters.

In this paper, we view the weight vectors as a matrix $\boldsymbol{M}$, which has N entries. For the entire network, we have H_D such matrices. Then for each matrix $\boldsymbol{M}$, we use the intensities of N related QoS features of the traffic to reflect the N connection. By extracting an abstract feature for each hidden unit, some deeper features of the traffic can be acquired. Assuming that the overall QoS requirements can be represented by a vector consisting of the corresponding QoS parameters, which can be represented by a real number that is bounded in the range of [0, 1] by processing. We specify the ith training sample $x^{(i)} \in R^d$, where i is an integer that represents the index of traffic, and R^d represents an N-dimensional real number Euclidian space, which consists of N QoS parameters. The set of training samples can be written as follows:

$$\boldsymbol{x} = \{x^{(1)}, x^{(2)}, x^{(3)}, \ldots\}. \tag{2}$$

Denoting $F(\cdot)$ as a feature-extracting function, each input $x^{(i)}$ from the set of training samples can be encoded into a hidden feature vector $F(\boldsymbol{x})$ based on the encoder activation functions [25], as follows:

$$F(\boldsymbol{x}) = F(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}) = \frac{1}{1 + \exp(-(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b})}, \tag{3}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{b}$ represent an encoder weight matrix and an encoder bias vector, respectively. Then, the hidden representation $F(\boldsymbol{x})$ is mapped from the hidden feature space back into a reconstruction $G(F(\boldsymbol{x}))$ of the input space based on the decoder activation functions, as follows:

$$G(F(\boldsymbol{x})) = G(\boldsymbol{W}_2F(\boldsymbol{x}) + \boldsymbol{c}) = \frac{1}{1 + \exp(-(\boldsymbol{W}_2F(\boldsymbol{x}) + \boldsymbol{c})}, \tag{4}$$

where $\boldsymbol{W}_2$ and $\boldsymbol{c}$ denote a decoder weight matrix and a decoder bias vector, respectively.

By attempting to reproduce the input of the model according to the lowest possible reconstruction error $L(x^{(i)}, G)$, the set of parameters of the encoder and mapping are learned simultaneously. In summary, the basic AE is a process to find a set of parameter vectors to minimise the reconstruction error, as follows:

$$P(\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{b}, \boldsymbol{c}) = \arg\min_{\{P\}} L(x^{(i)}, G(x^{(i)})) = \arg\min_{\{P\}} \frac{1}{2}\sum_{i=1}^{N} \| x^{(i)} - G(x^{(i)}) \|^2 . \quad (5)$$

This optimisation problem (5) is usually solved with the stochastic gradient descent method. As a result, the model parameters can be obtained. Considering that sparsity usually exists in the QoS characteristics of multimedia [41], it seems that the autoencoder should achieve a good performance by considering the sparse representation of the hidden layer [25]. Here, we directly penalise the output of the hidden layer activations to obtain sparsity in the representation with the Kullback–Leibler (KL) divergence [25]. Through combining the objective function with a sparsity constraint, we can generate a new objective function, and the new problem can be solved by the back-propagation (BP) algorithm, as follows [25]:

$$\mathrm{SAO} = L(x^{(i)}, G(x^{(i)}) + \gamma \sum_{j=1}^{H_D} \left(\rho \log\left(\frac{\rho}{\hat{\rho}_j}\right) + (1-\rho)\log\left(\frac{1-\rho}{1-\hat{\rho}_j}\right)\right), \quad (6)$$

$$\rho \log\left(\frac{\rho}{\hat{\rho}_j}\right) + (1-\rho)\log\left(\frac{1-\rho}{1-\hat{\rho}_j}\right) = 0, \quad \text{if } \rho = \hat{\rho}_j, \quad (7)$$

where γ and H_D denote the weight and the number of hidden units, respectively, ρ is a probability and equals to a small value close to zero (set to 0.005 in this paper), and $\hat{\rho}_j$ represents the average activation of hidden unit j, defined as

$$\hat{\rho}_j = \frac{1}{N\sum_{i=1}^{N} F_j(x^{(i)})}. \quad (8)$$

4.2 Stacked AEs

An SAE model is constructed by stacking AEs, in which the input and hidden layers of AEs are stacked together in a layer-by-layer manner. This model is used to form deep related QoS characteristics of multimedia flow data. First, the AE maps inputs in the 0th layer to the inputs in the first layer. After training the first layer AE, we can train subsequent layers with the output of the previous layer. At last, the decoder of the last layer is discarded, and we obtain the weights between the former and last layers by incorporating the input-to-hidden parameters. When we implement the subsequent classifier as a NN that aims to generate its input, we can adopt a fine-tuning operator that adjusts the parameters throughout the entire network when the classifier is trained.

To enhance the learning process or reduce the training error caused by a very small number of training patterns, we improve the architecture of the AE based on [42]. By stacking hierarchically multiple improved AEs, a deep network is thus created, which is illustrated in Fig. 6.

4.3 Classifying with multimedia related QoS characteristics

It is a challenge to extract the potentially related QoS features, since the multimedia traffic traces belong to the same class and often exhibit different related QoS characteristics under different network conditions. To handle the multimedia big data, we propose to utilise the DL approach to extract the inherent invariant characteristics for effective classification. In this paper, we firstly use an SAE to obtain the QoS characteristics, where the deep network is trained with the BP algorithm, and then we accomplish the classification by constructing a logistic regression classifier at the top layer.

In particular, we can obtain different levels of deep characteristics by adopting a different number of layers. Based on the work in [14, 23, 25], the training procedure is designed and described as follows:

Step 1: By solving the optimisation problem with objective function (6), the first layer is trained as an AE, where the inputs are the training samples.
Step 2: As the former step, the second layer is trained as an AE, for which its inputs directly come from the output of the first layer.
Step 3: The above procedure in Step 2 is repeated, until the last layer is trained.
Step 4: The input to the classification layer consists of the output from the last layer. Then its parameters are randomly initialised.
Step 5: Based on the BP algorithm, the parameters of all the layers are adjusted in a supervised manner.

The procedure of the proposed algorithm is stated as follows:

Step 1: Initialise the related parameters, including the desired number of hidden layers, the weight matrices, the bias vectors and so on.
Step 2: Execute the operations to pre-train the SAE, including training the hidden layers operator, while the inputs of the first hidden layer are the training samples. The inputs of the (k+1)th hidden layer are the outputs of the kth ($k > 1$) hidden layer. Through solving the optimisation problem, we can obtain the encoding parameters for the (k+1)th hidden layer.
Step 3: The fine-tuning stage: The encoding parameters are initialised randomly, and then the BP algorithm is utilised to fine tune the parameters throughout the entire network in a top–down fashion.

The flowchart of the proposed QoS class classification is described in Fig. 7. First, we capture the multimedia flow data from a live network as raw data. Through analysing the raw data, we can generate the related QoS features. By selecting features and operating normalisation, we can acquire the related QoS characteristic vectors. After obtaining the input dataset, we setup and train a SDAE, including pre-training and generating the AE network. When the training stage is over, we operate the fine-tuning procedure, where we use the SDAE to initialise and train a feed forward NN. To utilise the SAE network for multimedia traffic classification, a logistic regression layer is added at the top layer. Therefore, a deep architecture model is generated by combining the SAEs with the logistic regression layer for multimedia traffic classification. Finally, the output result is generated.

The performance of the proposed DL-based QoS class classification framework will be evaluated with real multimedia

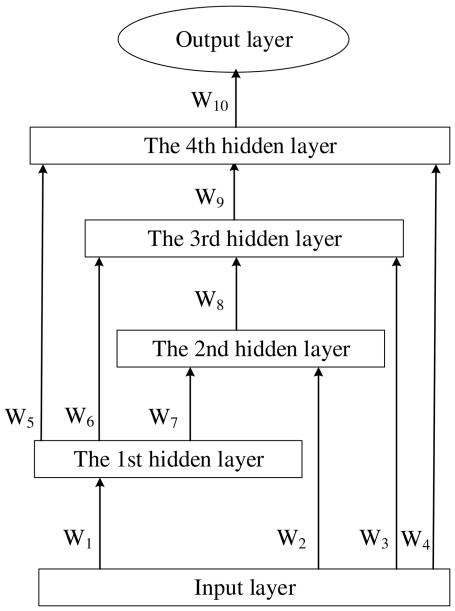

Fig. 6 *Deep network architecture and the improved layer-wise training of SAEs*

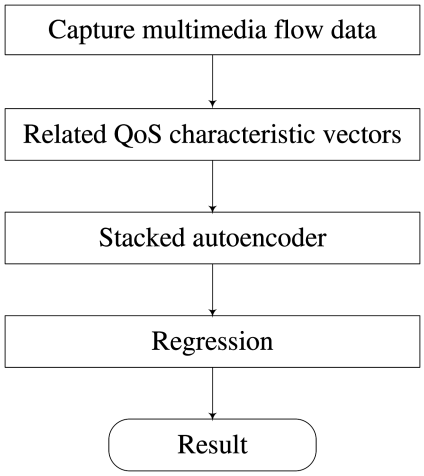

Fig. 7 *Flowchart of the proposed QoS class classification scheme*

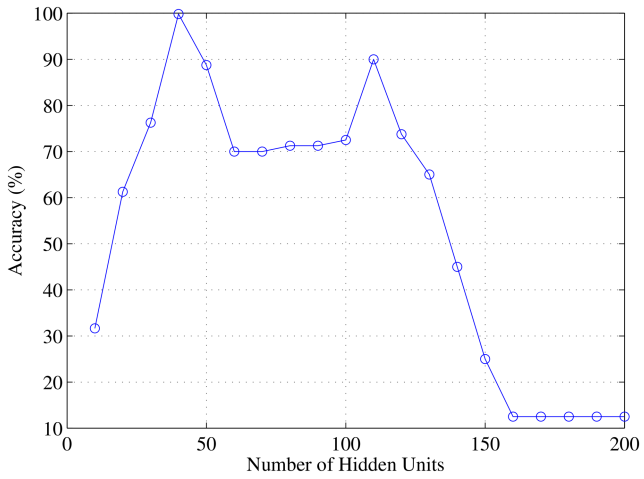

Fig. 8 *Impact of the number of hidden units in terms of accuracy*

datasets captured from a campus network in the next section, along with comparison with several benchmark schemes.

5 Experimental study

In this paper, the proposed approach is applied to the dataset collected from the campus of Anhui Normal University's Wuhu, China campus in the experiments, which is comprised with eight natural categories of multimedia traffic as shown in Table 2. Each of traffic trace is captured from a 35-min traffic flow. In the experiments, the dataset is divided into two parts: the first part is adopted as the training set, and the second part is used as the testing set. To obtain the best architecture of an SAE network, we perform a grid search runs in the experiment, where the hidden layer size is chosen from 1 to 4, and the number of hidden units is chosen from 10 to 200. Eventually we obtained the best architecture configuration, which is presented in Table 3.

For QoS feature classification, our best architecture consists of four hidden layers, and the number of hidden units in each hidden layer is 40. Our results indicate that the number of hidden layers should be neither too large nor too small. For a statistically accurate evaluation, the experiments are conducted over 50 runs. Considering that the number of hidden units determines the quality of QoS features, we perform the experiments with different numbers of hidden units while keeping the other parameter settings fixed. The experimental results are presented in Fig. 8.

In Fig. 8, it can be seen that the number of hidden units has a considerable impact on classification accuracy. When the number of hidden units is 40, the accuracy of the proposed method is up to 99.75%. However, when the number of hidden units grows to 160, the accuracy of the proposed method shows a large drop to 12.5%. In fact, other parameters (i.e. the number of hidden layers and the batchsize) also affect the performance of the proposed method.

Table 2 Components of the experimental dataset

Traffic type	No. traffic flows	Data amount, GB
video based on http	150	3.11
PPstream video	150	7.37
QQ video	150	7.28
sopcast video	150	7.38
CCTV online video	150	3.09
Xunlei video	150	11.09
Youku standard-definition video	150	1.83
Youku high-definition video	150	3.52

Table 3 Parameters used for running experiments in the structure of an SAE network

Parameter	Value
number of hidden layers	4
number of units in the hidden layers	[40, 40, 40, 40]
batchsize	1000
total epoches	400

Through extensive experimental studies, we select the parameters for the proposed method, which are given in Table 3. We find that the performance of the proposed method stops to improve after four layers with 40 hidden units due to overfitting. There is a serious issue concerning AEs in that if the hidden units are the same size or greater than the input units, an AE could potentially learn the identity function and become useless (e.g. just by copying the input) [39].

To comprehensively and quantitatively evaluate the performance of our method, five performance indexes are used for performance measurement in this paper, which are (i) the overall accuracy (OA), (ii) accuracy (A_c) [23], (iii) recall (R), (iv) precision (P), and (v) F1-measure(F_1). These performance metrics are defined as

$$\mathrm{OA} = \frac{\sum_{i=1}^{K} T_i}{\sum_{i=1}^{K} S_i} \tag{9}$$

$$A_c = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \tag{10}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{12}$$

$$F_1 = \frac{2 \times P \times R}{P + R}, \tag{13}$$

where K denotes the number of types of multimedia traffics, T_i represents the number of multimedia traffics that belong to the type i traffic and are correctly identified by the classifier, S_i is the number of traffics that belong to the type i traffic in the dataset, TP (true positive) denotes the number of traffics that are identified correctly by the classifier and are the correct traffics indeed, FP (false positive) represents the number of traffics that are identified correctly by the classifier and are the incorrect samples indeed, TN (true negative) denotes the number of traffics that are identified incorrectly by the classifier but are the incorrect traffics indeed, FN (false negative) denotes the number of traffics that are identified incorrectly by the classifier and are the correct samples in fact.

The performance of the proposed method is compared with six existing methods, including (i) the BP neural networks (denoted as BP NNs), (ii) the K-nearest neighbour (denoted as K-NN) [43], (iii) the random walk (denoted as RW) forecast method, (iv) the Naive Bayes method, (v) the radial basis function (denoted as

Table 4 Comparison of traffic classification methods in terms of OA

Classification method	Proposed	BP NN	K-NN	RW	N. Bayes	RBF NN	SVM
OA, %	99.75	96.86	92.1	95.34	88.86	94.63	96.16
OA gain, %	—	2.89	7.65	4.41	10.89	5.12	3.59

Table 5 Comparison of traffic classification methods in terms of accuracy (%)

Classification method	Video based on http	PPstream video	QQ video	Sopcast video	CCTV online video	Xunlei video	Youku video (SD)	Youku video (HD)
proposed	99.971	99.941	99.971	99.961	99.946	99.961	99.951	99.950
BP NN	99.453	98.957	99.498	99.130	99.050	88.667	99.034	99.111
K-NN	98.469	97.556	98.259	98.181	97.621	97.620	97.755	97.768
RW	99.342	98.508	99.201	98.760	98.322	98.871	98.634	98.560
Naive Bayes	97.567	96.537	97.635	97.057	96.606	96.617	96.792	96.900
RBF NN	99.169	98.407	99.049	98.667	98.327	98.919	98.462	98.476
SVM	99.462	98.764	99.391	98.961	98.704	99.194	98.817	98.798

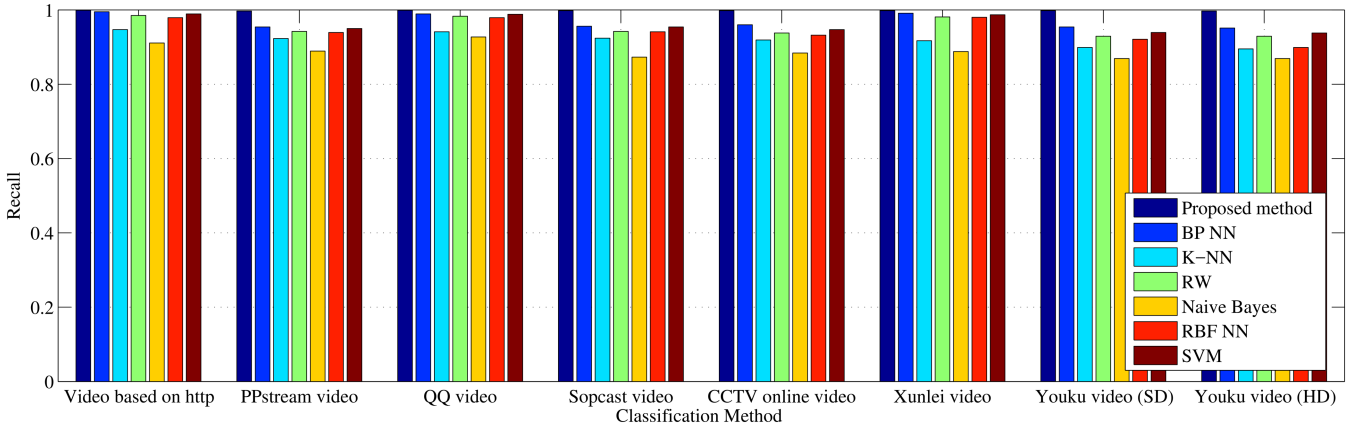

Fig. 9 *Comparison of traffic classification methods in term of recall*

RBF) NN model, and (vi) the SVM method [44], in terms of OA, accuracy [23], precision, recall, and F1-measure. Different from the existing works [43, 44], we focus on the new research problem in this paper, which is to utilise the DL method and the multimedia QoS characteristics to effectively classify the multimedia traffic big data. The experimental results are presented in Table 4, which are conducted over 50 runs in our experiments.

We compared the overall accuracy (i.e. OA) of the proposed method with BP NN, K-NN, RW, Naive Bayes, RBF NN, and SVM methods in Table 4. Among these seven competing methods, our proposed method achieves statistically the highest overall accuracy of 99.75%. The overall accuracy of BP NN, K-NN, RW, Naive Bayes, RBF NN, and SVM are 96.86, 92.1, 95.34, 88.86, 94.63, and 96.16%, respectively. The reason for this phenomenon is that small differences between the QoS features and QoS requirements can cause a bigger relative error when the number of traffic flows is small. Since the depth of QoS features has an impact on classifying multimedia traffic data, it seems that the method with deeper features can often easily achieve higher classification overall accuracies than methods with shallow features. Especially, we are more focused on classification results with a limited number of traffic flows in fact. Our proposed method can utilise the deep architectures (i.e. multiple-layer architectures) to discover the intrinsic QoS features at different levels, and can acquire a huge amounts of potential structures in QoS features. Like other DL algorithms, our proposed method can potentially generate progressively more complex and abstract features at higher layers, which are generally invariant when the input is, changed locally [23]. The proposed method can also represent traffic QoS features without prior knowledge [25], and it outperforms all other six classification methods with gains ranging from 2.89% (over BP NN) to 10.89% (over Naive Bayes).

The recognition performance of K-NN is influenced by the training set samples in the classification process, since it only calculates the 'nearest' neighbour samples. The RW method utilises a simple baseline to classify traffic, so that it has a poor performance in classifying traffic with complex QoS features. The training dataset has a considerable impact on BP NN, RBF NN, Naive Bayes, and SVM in the learning stage, which are dependent on the specific characteristics and have different recognition effects for different traffic types. We find the NN and SVM methods can achieve a pretty good performance for multimedia traffic classification. In Table 5, we can see that the accuracy of our proposed method is over 99.9% for all the eight types of traffics. Hence, our proposed method is highly effective for classifying multimedia traffic according to their QoS requirements.

Fig. 9 provides a visual display of the performance with respect to recall (i.e. R) achieved with the proposed method, BP NN, K-NN, RW, Naive Bayes, RBF NN, and SVM. It can be seen that the proposed method outperforms all the other six existing methods in term of recall. Note that the same observation is made with respect to all the other performance metrics. Each recall value for each method in the table is the average of 50 experiments. The proposed method achieves a better recall performance when compared with BP NN, K-NN, RW, Naive Bayes, RBF NN, and SVM, and is promising and effective with respect to recall performance. In terms of classification accuracy, BP NN and RBFNN's performance stop to improve after three layers due to overfitting, while the proposed method keeps improving as the number of layers is increased till as deep as we tested.

In Table 6, it is shown that our proposed method outperforms all the other six methods; it achieves the highest value in terms of precision. Our proposed method is also highly stable in term of precision, with precision values ranging from 0.99750 to 0.99870 or so. The BP NN method has precision values ranging from 0.95979 to 0.99849 or so. For the K-NN, RW, Naive Bayes, RBF NN, and SVM methods, the precision value has big drops under different traffic types. The maximum precision improvement of our proposed method is 0.03 over BP NN, 0.10 over K-NN, 0.06 over

Table 6 Comparison of traffic classification methods in terms of precision

Classification method	Video based on http	PPstream video	QQ video	Sopcast video	CCTV online video	Xunlei video	Youku video (SD)	Youku video (HD)
proposed	0.9980	0.9979	0.9985	0.9987	0.9975	0.9982	0.9985	0.9985
BP NN	0.9634	0.9644	0.9725	0.9752	0.9651	0.9598	0.9703	0.9985
K-NN	0.9386	0.8953	0.9289	0.9381	0.9023	0.9044	0.9290	0.9336
RW	0.9655	0.9429	0.9567	0.9617	0.9327	0.9354	0.9636	0.9586
Naive Bayes	0.9118	0.8605	0.9033	0.9083	0.8688	0.8666	0.8921	0.9006
RBF NN	0.9579	0.9384	0.9500	0.9555	0.9381	0.9394	0.9587	0.9602
SVM	0.9697	0.9533	0.9660	0.9647	0.9519	0.9521	0.9678	0.9676

Table 7 Comparison of traffic classification methods in terms of F1-measure

Classification method	Video based on http	PPstream video	QQ video	Sopcast video	CCTV online video	Xunlei video	Youku video (SD)	Youku video (HD)
proposed	0.9989	0.9976	0.9989	0.9984	0.9978	0.9979	0.9980	0.9980
BP NN	0.9790	0.9589	0.9806	0.9657	0.9628	0.9749	0.9619	0.9740
K-NN	0.9429	0.9090	0.9351	0.9311	0.9110	0.9107	0.9139	0.9140
RW	0.9750	0.9422	0.9697	0.9516	0.9351	0.9575	0.9463	0.9434
Naive Bayes	0.9113	0.8746	0.9150	0.8900	0.8762	0.8771	0.8807	0.8842
RBF NN	0.9686	0.9386	0.9641	0.9483	0.9442	0.9594	0.9395	0.9283
SVM	0.9794	0.9518	0.9767	0.9594	0.9496	0.9693	0.9533	0.9525

RW, 0.1 over Naive Bayes, 0.05 over RBF NN, and 0.04 over SVM.

The exact F1-measure values achieved by the seven schemes are given in Table 7, each being the average of 50 experiments. From Table 7, we also find that our proposed method is more effective than BP NN, K-NN, RW, Naive Bayes, RBF NN, and SVM in terms of F1-measure performance. Our proposed method achieves higher F1-measure values than other methods. The BP NN F1-measure values are range from 0.95891 to 0.98062; the K-NN F1-measure values are from 0.90902 to 0.94291; the RW values are from 0.93507 to 0.9750; the Naive Bayes values are from 0.87458 to 0.91502; the RBF NN values are from 0.92830 to 0.96855; and the SVM values are from 0.95178 to 0.97941. The maximum F1-measure performance improvement achieved by our proposed method is up to 0.03869 over BP NN, up to 0.08858 over K-NN, up to 0.06268 over RW, up to 0.12302 over Naive Bayes, up to 0.0697 over RBF NN, and up to 0.04816 over SVM. Our proposed method obviously achieves better F1-measure performance when compared with BP NN, K-NN, RW, Naive Bayes, RBF NN, and SVM. Thus, the effectiveness of the proposed method for multimedia traffic classification is validated with the experiments.

6 Conclusions

In this paper, we presented a DL-based approach with an improved SAE model for multimedia traffic classification. Different from the existing methods that only consider the shallow feature structure, since they are affected by limited training data, our proposed method can effectively extract the abstract, inherent QoS feature structure in multimedia big data, such as the invariant deeper features. We designed an SAE architecture, and trained the deep network with an improved model to avoid the impact from small amount of training data, and incorporated the fine-tuning process to optimise the model's parameters to enhance the classification performance. The proposed scheme was compared with six representative existing methods, including BP NN, K-NN, RW, Naive Bayes, RBF NN, and SVM, with traffic traces captured from a campus network. It is shown to outperform the benchmark schemes with respect to classification performance metrics including overall accuracy, accuracy, recall, precision, and F1-measure.

For future work, it is necessary to develop further studies with regard to the development of DL methods and multimedia big data. We shall utilise other DL models for IoT multimedia traffics classification from QoS perspective, and to study further these methods on different traffic datasets to evaluate their effectiveness.

7 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (grant no. 61401004), the Open Fund of the Key Laboratory of Grain Information Processing and Control (under grant no. KFJJ-2018-205), the NSF under Grant ACI-1642133, and by the Wireless Engineering Research and Education Center (WEREC) at Auburn University.

8 References

[1] Wang, W., Wang, Q.: 'Price the QoE, not the data: SMP-economic resource allocation in wireless multimedia Internet of Things', *IEEE Commun. Mag.*, 2018, **56**, (9), pp. 74–79

[2] Li, M., Si, P., Zhang, Y.: 'Delay-tolerant data traffic to software-defined vehicular networks with mobile edge computing in smart city', *IEEE Trans. Veh. Technol.*, 2018, **PP**, (1), pp. 1–14

[3] Karaadi, A., Sun, L., Mkwawa, I.-H.: 'Multimedia communications in Internet of Things QoT or QoE?'. Proc. 2017 IEEE iThings and IEEE GreenCom and IEEE CPSCom and IEEE SmartData, Exeter, UK, June 2017, pp. 23–29

[4] Al-Shammari, B.K.J., Al-Aboody, N., Al-Raweshidy, H.S.: 'Iot traffic management and integration in the QoS supported network', *IEEE Internet Things*, 2018, **5**, (1), pp. 352–370

[5] He, X., Wang, K., Huang, H., *et al.*: 'Green resource allocation based on deep reinforcement learning in content-centric IoT', *IEEE Trans. Emerg. Top. Comput.*, 2018, **PP**, (1), pp. 1–1

[6] Jalal, L., Anedda, M., Popescu, V., *et al.*: 'Qoe assessment for IoT-based multi sensorial media broadcasting', *IEEE Trans. Broadcast.*, 2018, **64**, (2), pp. 552–560

[7] He, Z., Mao, S., Jiang, T.: 'A survey of QoE driven video streaming over cognitive radio networks', *IEEE Netw.*, 2015, **29**, (6), pp. 20–25

[8] He, Z., Mao, S., Kompella, S.: 'Quality of experience driven multi-user video streaming in cellular cognitive radio networks with single channel access', *IEEE Trans. Multimedia*, 2016, **18**, (7), pp. 1401–1413

[9] Huang, Y., Mao, S., Midkiff, S.F.: 'A control-theoretic approach to rate control for streaming videos', *IEEE Trans. Multimedia*, 2009, **11**, (6), pp. 1072–1081

[10] Mao, S., Bushmitch, D., Narayanan, S., *et al.*: 'MRTP: A multi-flow real-time transport protocol for *ad hoc* networks', *IEEE Trans. Multimedia*, 2006, **8**, (2), pp. 356–369

[11] Wang, Z., Mao, S., Tang, P., *et al.*: 'A survey of multimedia big data', *IEEE/CIC China Commun.*, 2018, **15**, (1), pp. 155–176

[12] Chen, M., Mao, S., Zhang, Y., *et al.*: '*Big data: related technologies, challenges and future prospects*'. Springer Briefs Series on Wireless Communications (Springer, New York, NY, 2014)

[13] Chen, M., Mao, S., Liu, Y.: 'Big data: a survey', *Mob. Netw. Appl. (MONET)*, 2014, **19**, (2), pp. 171–209

[14] Chen, X.-W., Lin, X.: 'Big data deep learning: challenges and perspectives', *IEEE Access*, 2014, **2**, (1), pp. 514–525

[15] Wang, X., Gao, L., Mao, S.: 'Biloc: Bi-modality deep learning for indoor localization with 5 GHz commodity Wi-Fi', *IEEE Access*, 2017, **5**, (1), pp. 4209–4220

[16] Wang, X., Gao, L., Mao, S., *et al.*: 'CSI-based fingerprinting for indoor localization: a deep learning approach', *IEEE Trans. Veh. Technol.*, 2017, **66**, (1), pp. 763–776

[17] Wang, X., Gao, L., Mao, S.: 'CSI phase fingerprinting for indoor localization with a deep learning approach', *IEEE Internet Things*, 2016, **3**, (6), pp. 1113–1123
[18] Wang, X., Wang, X., Mao, S.: 'Cifi: deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi'. Proc. IEEE ICC 2017, Paris, France, May 2017, pp. 1–6
[19] Wang, X., Wang, X., Mao, S.: 'Deep convolutional neural networks for indoor localization with CSI images', *IEEE Trans. Netw. Sci. Eng.*, 2018, **5**, DOI: 10.1109/TNSE.2018.2871165
[20] Alam, M.R., Bennamoun, M., Togneri, R., *et al.*: 'A joint deep Boltzmann Machine (jDBM) model for person identification using mobile phone data', *IEEE Trans. Multimedia*, 2017, **19**, (2), pp. 317–326
[21] MÃijller, H., Unay, D.: 'Retrieval from and understanding of large-scale multi-modal medical datasets: a review', *IEEE Trans. Multimedia*, 2017, **19**, (9), pp. 2093–2104
[22] Weng, R., Lu, J., Tan, Y.-P., *et al.*: 'Learning cascaded deep auto-encoder networks for face alignment', *IEEE Trans. Multimedia*, 2016, **18**, (10), pp. 2066–2078
[23] Chen, Y., Lin, Z., Zhao, X., *et al.*: 'Deep learning-based classification of hyperspectral data', *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, 2014, **7**, (6), pp. 2094–2107
[24] Song, W., Tjondronegoro, D.W.: 'Acceptability-based QoE models for mobile video', *IEEE Trans. Multimedia*, 2014, **16**, (3), pp. 738–750
[25] Lv, Y., Duan, Y., Kang, W., *et al.*: 'Traffic flow prediction with big data: a deep learning approach', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (2), pp. 865–873
[26] Zhang, Y., Chen, M., Mao, S., *et al.*: 'CAP: crowd activity prediction based on big data analysis', *IEEE Netw.*, 2014, **28**, (4), pp. 52–57
[27] Wang, X., Wang, X., Mao, S.: 'RF sensing for internet of things: a general deep learning framework', *IEEE Commun.*, 2018, **56**, (9), pp. 62–69
[28] Sun, Y., Peng, M., Zhou, Y., *et al.*: 'Application of machine learning in wireless networks: Key technologies and open issues', under review
[29] Zhu, W., Cui, P., Wang, Z., *et al.*: 'Multimedia big data computing', *IEEE Multimedia*, 2015, **22**, (3), pp. 96–106
[30] Zhang, J., Xu, C., Jing, P., *et al.*: 'A tensor-driven temporal correlation model for video sequence classification', *IEEE Signal Process. Lett.*, 2016, **23**, (9), pp. 1246–1249
[31] Fan, R., Zhang, Y., Li, B.: 'Motion classification-based fast motion estimation for high-efficiency video coding', *IEEE Trans. Multimedia*, 2017, **19**, (5), pp. 893–907
[32] Dubin, R., Dvir, A., Pele, O., *et al.*: 'I know what you saw last minute-encrypted HTTP adaptive video streaming title classification', *IEEE Trans. Inf. Forensics Security*, 2017, **12**, (12), pp. 3039–3049
[33] Huang, W., Song, G., Hong, H., *et al.*: 'Deep architecture for traffic flow prediction: deep belief networks with multitask learning', *IEEE Trans. Intell. Transp. Syst.*, 2014, **15**, (5), pp. 2191–2201
[34] Hou, W., Gao, X., Tao, D., *et al.*: 'Blind image quality assessment via deep learning', *IEEE Trans. Neural Netw. Learn. Syst.*, 2015, **26**, (6), pp. 1275–1286
[35] Nie, L., Wang, M., Zhang, L., *et al.*: 'Disease inference from health-related questions via sparse deep learning', *IEEE Trans. Knowledge Data Eng.*, 2015, **27**, (8), pp. 2107–2119
[36] Zhao, X., Li, X., Zhang, Z.: 'Multimedia retrieval via deep learning to rank', *IEEE Signal Process. Lett.*, 2015, **22**, (9), pp. 1487–1491
[37] Kereliuk, C., Sturm, B.L., Larsen, J.: 'Deep learning and music adversaries', *IEEE Trans. Multimedia*, 2015, **17**, (11), pp. 2059–2071
[38] Bengio, Y., Courville, A., Vincent, P.: 'Representation learning: A review and new perspectives', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, **35**, (8), pp. 1798–1828
[39] Yang, H.-F., Dillon, T.S., Chen, Y.-P.P.: 'Optimized structure of the traffic flow forecasting model with a deep learning approach', *IEEE Trans. Neural Netw. Learn. Syst.*, 2017, **28**, (10), pp. 2371–2381
[40] Wireshark: 'Wireshark–Go deep', Available at https://www.wireshark.org/
[41] Hu, Y., Peng, Q., Hu, X., *et al.*: 'Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering', *IEEE Trans. Services Comput.*, 2015, **8**, (5), pp. 782–794
[42] Wilamowski, B.M.: 'How to not get frustrated with neural networks'. Proc. 2011 IEEE Int. Conf. on Industrial Technology (ICIT), Auburn, AL, March 2011, pp. 5–11
[43] Dibeklioglu, H., Salah, A.A., Gevers, T.: 'Recognition of genuine smiles', *IEEE Trans. Multimedia*, 2015, **17**, (3), pp. 279–294
[44] He, J., Yang, Y., Qiao, Y., *et al.*: 'Accurate classification of P2P traffic by clustering flows', *IEEE/CIC China Commun.*, 2013, **10**, (11), pp. 42–51